LEVEL II

12

AD A092293

TECHNICAL REPORT 80-05

Final Technical Report:

# The Design, Development, Demonstration, and Transfer of Advanced Command and Control (C2) Computer-Based Systems

James F. Wittmeyer, III





CSM INC.

Computer Systems Management, Inc.
1300 WILSON BOULEVARD, SUITE 102 • ARLINGTON, VIRGINIA 22209

(14) CSM-

(12)

TECHNICAL REPORT 80-05



# Final Technical Report:

(6) The Design, Development, Demonstration and Transfer of Advanced Command and Control ($C^2$) Computer-Based Systems.

by

(10) James F. Wittmeyer, III

(9) Final rept.
5 Nov 79 - 30 Sep 80,

(11) 30 Sep 80

(12) 78

(15) MDA903-80-C-0155,
ARPA Order-3829

CSM INC.
Computer Systems Management, Inc.
1300 WILSON BOULEVARD, SUITE 102
ARLINGTON, VIRGINIA 22209



411586

TECHNICAL REPORT 80-05

# FINAL TECHNICAL REPORT:

# THE DESIGN, DEVELOPMENT, DEMONSTRATION AND TRANSFER OF ADVANCED COMMAND AND CONTROL ($C^2$) COMPUTER-BASED SYSTEMS

| | |
|---|---|
| ARPA Order No.: | 3829 |
| Contractor: | Computer Systems Management, Inc.<br>1300 Wilson Boulevard<br>Arlington, Virginia 22209 |
| Effective Date of Contract: | 11/5/79 |
| Contract Expiration Date: | 9/30/81 |
| Contract No.: | MDA903-80-C-0155 |
| Principle Investigator: | James F. Wittmeyer, III<br>(703) 525-8585 |
| Contract Period Covered: | 11/5/79 - 9/30/80 |
| Short Title of Work: | Advanced $C^2$ Computer-Based Systems |

This research was sponsored by the Defense Advanced Research Projects Agency under ARPA Order Number 3829; Contract Number MDA903-80-C-0155; and Monitored by DSS-W. The views and conclusions contained in this document are those of the author and should not be interpreted as necessarily representing the official policies, either express or implied, of the Defense Advanced Research Projects Agency or the United States Government.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

# REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

| | |
|---|---|
| 1. REPORT NUMBER: 80-05 | 2. GOVT ACCESSION NO.: AD-A092 293 |
| | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle): The Design, Development, Demonstration, and Transfer of Advanced $C^2$ Computer-Based Systems | 5. TYPE OF REPORT & PERIOD COVERED: Final Technical Report 11/5/79- 9/30/80 |
| | 6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s): James F. Wittmeyer, III | 8. CONTRACT OR GRANT NUMBER(s): MDA903-80-C-0155 |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS: Computer Systems Management, Inc. Suite 102 1300 Wilson Boulevard, Arlington, Va 22209 | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: ARPA Order No. 3829 |
| 11. CONTROLLING OFFICE NAME AND ADDRESS: DARPA 1400 Wilson Boulevard Arlington, Virginia 22209 | 12. REPORT DATE: 9/30/80 |
| | 13. NUMBER OF PAGES: 56 |
| 14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Defense Supply Service-Washington (DSS-W) The Pentagon Washington, D.C. | 15. SECURITY CLASS. (of this report): Unclassified |
| | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT (of this Report)

Recommended for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Recommended for public release; distribution unlimited

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Command and control ($C^2$); computer-based systems; systems development; systems demonstration; timesharing; minicomputers; microcomputers; information systems; decision systems; forecasting systems; training systems; readiness systems

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The conduct of research into the nature and use of advanced command and control ($C^2$) computer-based information, decision, forecasting, training and readiness systems requires efforts in the $C^2$ computer-based systems design, development, demonstration, transfer, and documentation areas. This report examines these areas and discusses a multiple PDP 11/70- and minicomputer-based research configuration.

## SUMMARY

This Final Technical Report covers the period from November 5, 1979 to September 30, 1980. The Tasks/Objectives and/or Purposes of the overall project are connected with the design, development, demonstration and transfer of advanced command and control ($C^2$) computer-based systems. The Technical Problems addressed include providing support for $C^2$ systems development in the information, decision, forecasting, training and readiness areas via computer timesharing and demonstrations and how to accelerate development through structured documentation. The General Methods employed have involved building and maintaining a multiple PDP 11/70 timesharing system and simultaneously configuring multiple microcomputer systems. In addition, problems have been approached via the structuring of a realistic demonstration and test environment. Technical Results to date include a seven day, twenty four hour a day design and $C^2$ development support system, the development of realistic testing and demonstration scenarios and capabilities, and the completion of a plan for the transfer documentation of $C^2$ system development. The Hardware Configuration utilized for this research is entirely GFE. The systems software includes the UNIX operating system, FORTRAN IV PLUS, C, and some limited statistical packages; the applications software consists of numerous programs in the $C^2$ information, decision, forecasting, training and readiness areas. Future Research, in FY81, will continue in the general area especially as it moves closer to the communications and man-computer relations areas.

THE DESIGN, DEVELOPMENT, DEMONSTRATION, AND TRANSFER OF ADVANCED COMMAND AND CONTROL ($C^2$) COMPUTER-BASED SYSTEMS

## TABLE OF CONTENTS

# FIGURES

# 1.0 INTRODUCTION

## 1.1 Problems

The Defense Advanced Research Projects Agency's Cybernetics Technology Division (DARPA/CTD) has as its primary mission the development, application, and transfer of computer-based systems for improved Department of Defense (DoD) information management and display, forecasting, decision making, communications, training and human performance, especially as such activities occur in Command and Control ($C^2$) environments. Unfortunately, however, the research and development (R&D) process connected with the development of advanced $C^2$ computer-based systems is fraught with problems. Specifically, such problems may be categorized as follows:

### 1.1.1 Computer-based Systems Design

1.1.1.1. Neglect of "front-end" analysis. This problem runs rampant throughout all of the projects that have as a final product either computer-based systems software or analytic or descriptive data sets. Moreover, front-end analysis has seldom been conducted in any of the areas of the computational needs of the hardware and software spectrum. An example illustrating the myriad of problems that can occur without proper design analysis is the current state of the Terrorism Research and Analysis Project (TRAP). TRAP

software is a TEKTRONIX 4052 resident BASIC program. Many demonstrations of its capabilities show the extreme value it has as a sophisticated operations research system. However, it is designed to run on a very specific type of graphics microprocessor for which there is only one manufacturer. It can also only be demonstrated on large screen projection via a Hughes scan converter and a specially modified 4052.

1.1.1.2 Expensive and fundamental "disconnects" among the programmer, the intended product, and the ultimate user. So often, through a very basic misunderstanding in the design phase, there occurs a strange phenomenon which seperates the intended purpose of the research tool from its preparation and construction. This "distance" often causes enormous problems in the final stages of software implementation and transfer. If in fact the product delivered is neither what was intended nor what can be used, it must be re-written, re-tested, re-validated and re-documented--generally a very expensive process. The cost in man-hours alone is sometimes staggering. Further costs of late-delivery, and other projects suffering because of a reshuffling of priorities are also not inconsequential. It is important to keep sight of who the ultimate user is, where the tool will be utilized, when it must be ready to be effective, and how it should be implemented.

1.1.1.3 Non-standardized data sets and codebooks. In the early years of the conceptualization and creation of the Demonstration and Development Facility (DDF), it became evident that a large portion of the DDF user community would be tasked with the creation and maintenance of various data sets. This function has no less importance than the analytic software tools which often evolve from such data sets. But here too we find design flaws. Care should be taken, at the outset, to standardize the coding and collection of the data, particularly with a view to how they might later be analyzed and processed via a computer-based delivery system.

### 1.1.2 Computer-based Systems Development

1.1.2.1 Hardware. Here selection and usage problems are often encountered in many of the more advanced research projects. Research products that use ineffective computer systems will fail no matter how high the value of the research product. Correct hardware selection is critical to the successful development and transfer of a research product. Factors such as portability, maintenance costs, backup systems, commercial availability, and life-expectancy are all important to the hardware/software marriage.

1.1.2.2 Software. Here, development problems arise throughout all programming activities in every organization. Poor software development procedures result in much wasted

time, effort and cost. Typical problems concentrate usually around language selection and implementation. Many languages are incompatible with one another. This problem of incompatibility exists not only between operating systems (such as UNIX vs. RSX11-M) but on computer systems such as HONEYWELL Level 6 vs. DEC 11/70 as well.

Pragmatic decisions must be made not only as to the user availability of languages but the fundamental choice of a language for an application. For example, if a short term project, with a specific transfer application requires APL, then it should be used. But, on the other hand, if the need for the resultant research tool is more universal, then it would be wrong to allow programming to begin in APL. For example, Decisions and Designs, Inc. and the Computer Corporation of America, Inc. constructed a very valuable set of decision making and evaluation aiding tools for the U.S. Marine Corps. The results were spectacular but the software was written in APL, and the Marines could only accept transfer of software written in COBOL.

Also under the jurisdiction of software development comes the very important topic of documentation. Throughout the development phase of either software tools or data sets, time <u>must</u> be spent on building effective documentation. Documentation takes the forms of internal documentation,

systems specifications, users manuals, and codebooks. Improperly documented code, or code without documentation, lives only as long as its author.

1.1.2.3 Coordination. Far too often redundant coding, or redundancy of effort exists im program development. The problem comes about partially through the "not invented here" syndrome. In other cases the "re-invented the wheel" syndrome applies.

1.1.3 Demonstrations

1.1.3.1 Opposing "cultures." Contractors typically by the nature of their roles as researchers are disconnected from the nature of organizational decision making. As a result, they are seldom able to provide the necessary context for an effective demonstration. The real value of a Demonstration and Development Facility (DDF)--cognizant of the critical distance between the researcher and the "user"--comes into play during demonstrations. Since contractors are generally unskilled in the art of interacting effectively with government, military and civilian personnel at all levels, they harbor latent misunderstandings and confusions about the nature of the operational world. All of this predicts to the failure on the part of the researcher to incorporate into his or her computer-based system the necessary user-oriented features so critical to generating interest, appeal and, ultimately,

acceptance. An anecdotal example recalls the first use of the Joint Chiefs of Staff (JCS) regional capability of the EWAMS for use by a military analyst, who promptly asked why he must relearn all the nations of the world in three letter WEIS representation rather than the SOP military two-letter designation.

1.1.3.2 Demonstration staging. The success or failure of a developed software product often completely depends upon the nature of the environment in which the product is first introduced to a potential user.

1.1.3.3 Premature demonstrations. Another problem all too often encountered is the premature release and demonstration of a product. Contractors by their nature have very little feeling for proper timing, and in this regard, often demonstiate too early with disastrous possibly even fatal results.

### 1.1.4 Transfer

1.1.4.1 Targeting. Too often we are not cognizant of the overall requirement that we are trying to address via the development of a computer-based system. That is to say, we must remain mindful of the target, need, use and problem to be solved, and avoid becomming overly impressed with techniques or inventive solutions, which although may be major break-

throughs in and of themselves, must address a targeted need.

1.1.4.2 <u>Transfer site requirements.</u> There exists a significant problem in the failure to adequately survey the hardware and software at a proposed user's site, often resulting in a serious mismatching. Transfer efforts must be done professionally and expertly. Those individuals responsible for the evaluation of the users hardware configuration and facilities have little room for error. Beginning with the design phase, all work should be aimed at effective solutions to specific development problems. The final phase in the solution of that problem must take into consideration every possible condition which could prevent the integration of the new work with the existing technology. Problems encountered in the transfer process are highly visible to the ultimate user and may overshadow the impression of competence as well as confidence in the delivered work.

1.1.4.3 <u>Documentation.</u> More commonly, there is a failure to provide necessary instructional documentation along with the software and data sets being transferred. This problem has the potential for being as dangerous as a faulty transfer. Without the proper documentation, interest in the product will wane. Because it is too difficult to understand, it will not be used.

1.1.4.4 Marketing. Another problem that can occur is the failure to adequately assess the bureaucratic back-drop necessary for thorough, formal transfer. While this is not to say that informal transfer is not valuable, it is to say that often without top-down approval from the "chain-of-command," transfer could be at best delayed, or at worst, prevented.

## 1.2 Proposed Solution

It has become clear that the research and development process especially as it appears to the development of advanced $C^2$ computer-based systems has many problematic areas. The cataloguing of these problems is the first step toward the framing of a solution. Step two requires the establishment of a set of general and specific mission-oriented objectives by which the solutions can be determined and applied. Some general and specific solutions and objectives developed in FY80 appear below.

Objective 1 - The establishment of a new and vital design phase for all candidate DARPA/CTD contractor software. A rigid set of design standards have been applied to all new and intended software products. Through the application of these standards the software and data requirements can be categorized into groups which will serve to indicate the needs of the project which are to be provided by DDF personnel. Further analysis is made at this time to determine the pre-existence

of data sets that may fulfill the requirements of the effort. As a function of this design phase, clear cut alternatives can be examined, weighted, and selected thereby insuring that the porposed effort is both well conceived and within the limits of acceptable programming practices.

Objective 2 - Provide a superstructure of effective hardware and software development capabilities. Every effort has been made to accommodate the hardware and software requirements of the DDF user community. This effort has been centered around the operation of DEC PDP 11/70 mainframe computer systems. The full resources of this service have been extended in an attempt to adequately support on-going development, design and transfer efforts. However, the provision of this service is not the only goal. Also included in the development phase are: hardware selection assistance, programming assistance, training and advice, creation of documentation standards and a concerted effort to keep all researchers informed of technical advances made both inside and outside the DARPA/CTD community.

Objective 3 - Take a leadership role in the organization and presentation of professional and effective demonstrations. The advanced computer-based systems and data developed for DARPA/CTD must endure intensive critique by a viewing audience. Therefore, it becomes increasingly important that a concerted

effort be made to establish and conduct policies for and training about effective demonstrations. Here too resides the need for physical as well as consultantive services. It is very important to provide a well organized program of demonstrations in an environment conductive to generating interest, appeal and hopefully acceptance of the research products being developed by those representatives of the operational world who may wish to adopt new computer-based systems.

Objective 4 - Provide expertise ready to address the problem of transferring selected software and data from research status to operational service. As part of the overall mission of DARPA/CTD successful research products must, after design, development and demonstration, be transferred to a proposed user's site. Care must be taken to adequately analyze the systems and facilities available at the transfer site so as to assure that the least amount of difficulty is encountered during the transfer. During the transfer phase all supporting documentation should be assembled to accompany the software and data, comprising a complete package which is useable and understandable. Also, care should be taken to coordinate with the transferee to insure that the products being delivered are what is expected as well as needed.

Objective 5 - Create a new management approach to the

organization of the DDF. Most, if not all, of the technical problems currently facing the DDF user-community cannot be resolved by technical adjustments alone. These problems require a new management model, a re-organization to accomodate technical advancement.

## 2.0 THE DESIGN & TRANSFER OF ADVANCED $C^2$ COMPUTER-BASED SYSTEMS

### 2.1 The Design, Development, Demonstration, Transfer and Documentation Tasks

During the course of Fiscal Years 1980 and 1981 all of the tasks associated with computer-based systems research will be addressed. After briefly addressing the context for our work--the DARPA/CTD FY80 research program--we will turn in section 2.1.2, to computer-based systems design and, in section 2.1.3, to computer-based systems transfer.

#### 2.1.1 FY80 DARPA/CTD Research Program

It must be noted that the DARPA/CTD research program is constantly evolvong. CSM is now working from one blueprint; at the same time we continually monitor changes in the nature and direction of the program in order to assess possible impact upon the operation of the DDF.

#### 2.1.2 Computer-Based System Design

One method for designing computer-based systems requires that the intended system pass through a set of filters as suggested below in Figure 1. Filter one asks whether the system is to be a research system or an application system. Research systems are generally aimed at developing techniques and ideas, so that they may be proved worthy. On the other hand, applications systems draw upon previous research in such

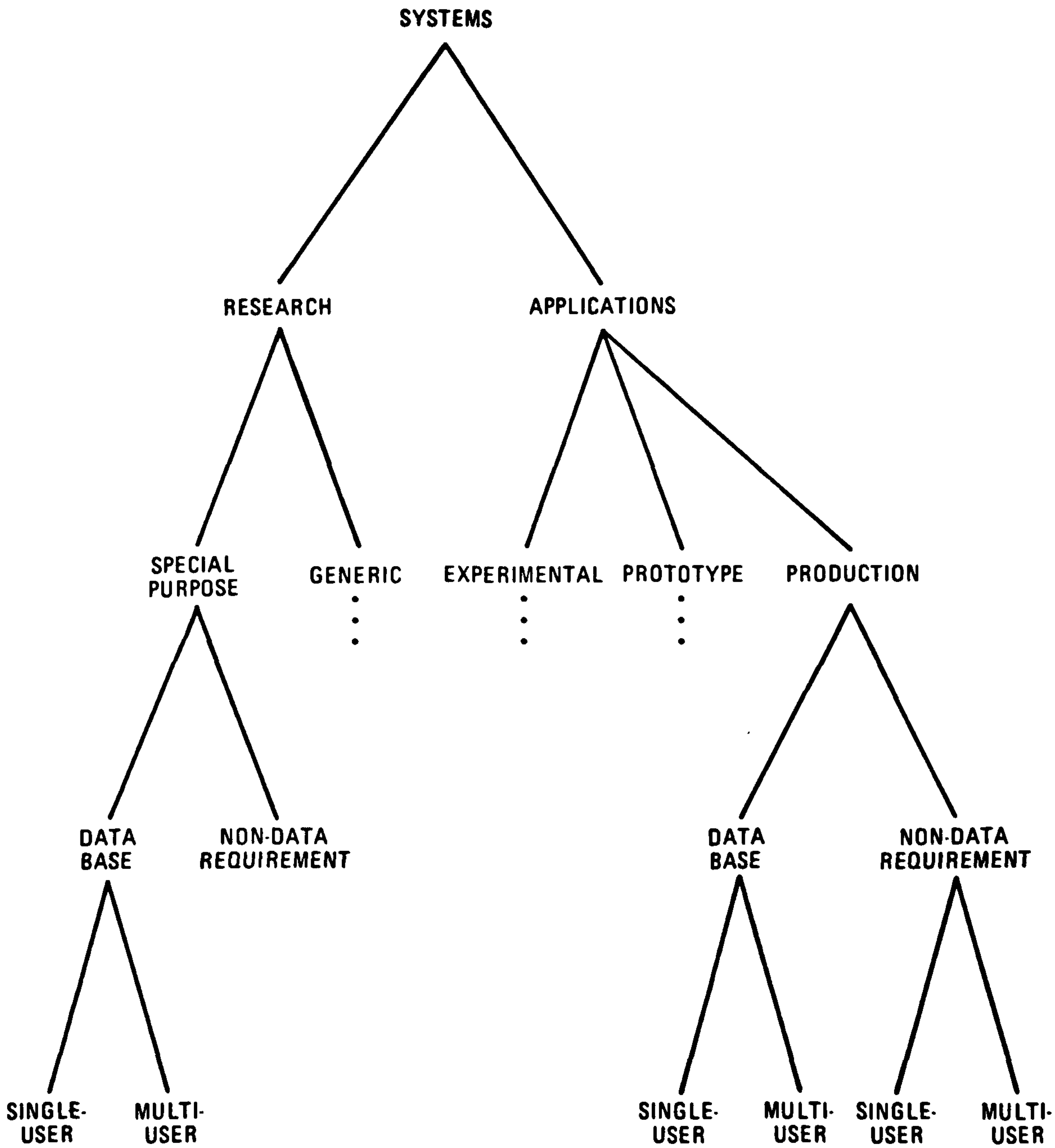


Figure 1
"DESIGN FILTERS" FOR DARPA/CTO COMPUTER-BASED SYSTEMS

a way as to expand previous ideas into complete experimental or working models.

The second set of filters further expands research systems into two categories: special purpose or generic. Special purpose differs from generic in the sense that it has an intended single purpose. Applications systems are subdivided into three types: (1) experimental, (2) prototype, and (3) production. A sequence in the maturization of an applications software system is first that of an experimental model, which thus leads to prototype development and finally full production model(s).

The third level tests the system for its data requirements; does it require a prestored data set or is the data to be generated on-line during execution of the system? An example of prestored data exists in the large WEIS data set required for execution of several modules of the EWAMS, whereas ADT systems elicit user probabilistic assessments which are then saved for further systems analysis. The last set of filters to test the user requirement asks the question: will the system be on-line to multi-users, or will it be a single-user (stand alone)? The answer to this question is as important in the design of a system as the other filters.

For example, had the EWAMS been passed through these filters the first level would have distinguished it as an

applications system. It would have been distinguished as prototype in nature, requiring a prestored data set, and multi-user-oriented. Unfortunately, these filters were not employed and numerous difficulties plagued the development, demonstration and transfer phases of EWAMS. Retrospectively, EWAMS might best have been designed as two systems employing the production versus prototype filters.

These filters, then, are illustrative of the kind that CSM attempts to develop and apply for DARPA/CTD at their direction as their computer-based systems development requirements arise.

2.1.2.1 <u>Disconnects.</u> The application of the above filters will reduce--if not eliminate--many of the "disconnects" discussed in section 1.1.1.2. It is our view that prudent passage through the above discussed design filters will minimize the "distance" among the research, the intended applications or research area, and the ultimate user.

2.1.2.2 <u>User Emphases.</u> Throughout the design process, the intended user or users should be studied carefully. At the lowest level of the design filtering process, then, particular attention must be paid and specific analyses should be performed. Such analyses include, but are not limited to:[1]

- Users

  - Their behavior in general; how to determine the properties of a particular user population; the implications of those properties for the interactive system;

- Tasks

  - What tasks users perform; how to determine tasks involved in an application;

- Requirements Analysis

  - How to analyze information requirements; how to select appropriate types of problem-solving, clerical and user support aids; allocation of basic tasks to user or computer; modeling of user-system interactions; evaluation of basic design;

- Interactive Dialogue

  - Properties of different dialogue types; selection of appropriate dialogue type(s); detailed design of command language, system access structures, tutorial aids, etc.;

- Output Devices and Techniques

  - Properties of display devices; implications of dialogue method for display device selection; selection or design of display device(s); detailed display design, formatting, coding techniques, etc.;

- Input Devices and Techniques

  - Properties of input devices; implications of dialogue methods for input device selection; selection or design of input device(s); and

- Evaluation of System Performance

  - Use of subjective evaluations, objective performance measures.

Users should thus be identified and classified as suggested below:

- Naive Users (Inexperienced with computers)

  - Computer-naive users are actually a very heterogeneous group, but have many common properties. Naive users benefit greatly from computer-initiated dialogue, usually require more tutorial features. Correct implicit "mental model" of computer systems and interactive dialogue cannot be assumed, must be explicitly conveyed by system. Naive user population has many detailed implications for dialogue design. Smooth transition from naive to experienced user is often difficult in current systems.

- Managers (Including Military Commanders, etc.)

  - Managers tend to have highly variable information needs; current systems are often too rigidly constraining to satisfy those needs. Managers tend to place high negative value on own effort, have considerable discretion with respect to mode of system use or nonuse. Thus, very low "impedance" is required to capture manager as direct user. If dissatisfied, manager tends to resort to "distant use" (interposing operator between manager and system) or partial use.

- Scientific and Technical

  - High proportion report dissatisfaction with available automated tools. These users often respond to such dissatisfaction by becoming personally involved in design or implementation of software tools, or by altering task to match available tools.

Tasks as well should be specified ideally in taxonomy form.

User requirements analyses should preceed any and all implementation. Some requirements analysis techniques appear below:

- Use of questionnaires to obtain ratings of the relative importance of various categories of information and system features;

- Use of questionnaires to obtain estimates of time spent on each task associated with recipient's job;

- "Repertory Grid Technique", a questionnaire-based technique for determining user's "cognitive frame of reference";

- "Delphi Technique", a survey technique in which recipient's responses are fed back, anonymously. Recipient responds again, while aware of previous responses of entire group;

- "Policy Capture", one of several techniques for developing quantitative relationships between perceived system desirability and specific system features. In this

case, relationship takes the form of a multiple-regression equation;

- Interviews with users to determine information requirements, decision points, organizational constraints, etc;

- "Ad Hoc Working Group", in which subject-matter experts devise system requirements by analysis and negotiation;

- "Critical Incident Technique", in which users are asked, via interview or survey, for information about incidents of particular success or failure in the process of which the computer system will be a part;

- Job analysis techniques, such as task analysis, link analysis, and activity analysis, which attempt to characterize user behavior on the basis of direct observation;

- "Paper" simulation, in which the possible function of a computer system is simulated by human observers, in order to obtain information about the user's problem-solving and information-seeking behavior;

- "Protocol analysis", in which the user comments extensively on his activities during simulated problem solving, and formal content analysis of the resulting commentary ("protocol") is used to make inferences about user behavior and problem-solving processes; and

- Interactive simulation or gaming, in which the actual system, or an interactive computer simulation of the system, is used with a contrived scenario to observe user behavior and system performance.

The selection of <u>interactive dialogue technique</u> is also critical. Some properties of interactive dialogues appear below on an assumed scale:

- Initiative
  - Initiative is concerned with whether the user of the computer initiates the individual information transactions within the dialogue. If the computer asks questions, presents alternatives, etc., and the user responds, the dialogue is "computer-initiated". If the user inputs commands without such computer "prompting", the dialogue is "user-initiated". "Mixed initiative" and "variable-initiative" dialogues are also possible;

- Flexibility
  - Flexibility is a measure of the number of ways in which a user can accomplish a given function. High flexibility can be achieved by providing a large number of commands, by allowing the user to define or redefine commands, etc.;

- Complexity
  - Complexity is related to flexibility. Complexity is a measure of the number of options available to the user <u>at a given point</u> in the dialogue. <u>Low complexity</u> can be achieved by using few commands, or by partitioning the commands so that the user selects from a small set at any given time;

- Power
  - Power is the amount of work accomplished by the system in response to a single user command. In a dialogue with powerful commands, the user may

accomplish, with a single command, an operation which would require several commands in a system with less powerful commands. Power is related to flexibility and complexity;

- Information Load

  - Information load is a measure of the degree to which the interaction absorbs the memory and/or processing resources of the user.

- System Response Time

- Communication Medium

Types of interactive dialogue to be selected by the designer appear below:

- Question-and Answer

  - Computer asks a series of questions, to which user responds;

- Form-filling

  - Computer presents form with blanks. User fills in blanks;

- Menu Selection

  - Computer presents list of alternatives, and user selects one or more;

- Function Keys with Command Language

  - User indicates desired action by depressing keys, each of which represents a command, command modifier, or parameter value;

- User-initiated Command Language
  - User types commands, perhaps using mnemonic abbreviations;
- Query Language
  - User inputs questions or data-base access procedures to a data base system. System produces response or report;
- Natural Language
  - Dialogue is conducted in user's natural language (e.g., English); and
- Interactive Graphics
  - Generation of pictorial displays, ability of user to select displayed entities and spatial locations by pointing or similar nonverbal means.

The evaluation and selection of output devices is also critical. Some variations are presented below:

- Refreshed CRT
  - The ordinary, refreshed CRT is currently the "basic" computer display. A good deal of data exists concerning appropriate visual properties of CRT displays. Studies which have compared user performance using CRTs with performance on other display devices do not provide a satisfactory basis for selection decisions;
- Storage Tube CRT

- For some graphical applications, direct-view storage tubes may be preferable to refreshed displays. The storage tube allows very high-density, flicker-free displays, but imposes significant constraints on interactive dialogue. Although information exists concerning the basic functional advantages and disadvantages of such displays, no empirical data pertaining to human factors concerns were found;

- Plasma Panel Display

  - Plasma panel displays are inherently "dot", or punctate, displays, and studies of symbol generation method are relevant. Little empirical information exists on human performance aspects of plasma displays *per se*;

- Teletypewriter

  - Reasonable guidelines exist with respect to the design of teletypewriter terminals, including both physical and functional properties;

- Line Printer

  - Research on typography is voluminous and directly applicable. Research dealing directly with the line printer used in computer output is scanty, but consistent with findings of typographic research (e.g., mixed upper-lower case is best for reading comprehension). Guidelines are not known to exist, but could be constructed with additional survey of typographic research literature. Use of line printers for "pseudographic" displays is common, little discussed in the literature. Pseudographics is an inexpensive way to convey simple graphical information, and should probably be used more widely in batch applications;

- Laser Displays

  - Reasonable human factors guidelines with respect to visual properties have been proposed, but these displays are not widely used;

- Tactile Displays

  - Although some tactile displays have been proposed or even developed, little human factors research has been done other than that concerned with prosthetics;

- Psychophysiological Displays

  - Psychophysiological input is technically feasible now, but psychophysiological displays are still only a topic for research; and

- Large-Screen Displays

  - There is conflicting evidence with respect to the performance effects of large-group vs individual displays. The main advantages of large-screen displays are a larger display area and the existence of a single display which is clearly the same for all viewers. Unfortunately, higher display content is not achievable due to the resolution limits of existing technology (e.g., light valve displays), and may be unachievable in principle, since the large-screen display usually subtends a smaller visual angle than an individual display located close to the user.

Input devices come next; options appear below:

- Keyboard;

- Lightpen, Lightgun;
- Joystick;
- Trackball;
- Mouse;
- Graphical Input Tablet;
- Touch Panel
- Knee Control
- Thumbwheels, Switches, Potentiometers;
- Tactile Input Devices;
- Psychophysiological Input Devices;
- Automated Speech Recognition;
- Hand Printing for Optical Character Recognition (or for Subsequent Entry by Typist);
- Mark Sensing;
- Punched Cards; and
- Touch-Tone Telephone.

Our design filters are thus extremely functional and higher order; the selections of detailed computer-based system components

--oriented to the user and actual use--are far more complex and critical to the successful development and use of advanced $C^2$ computer-based systems.

### 2.1.3 Computer-Based System Transfer

If adequate requirements analyses are performed, the transfer process can be greatly improved. Candidly, it is infrequently the case that requirements analyses are performed; instead, most computer-based systems are designed as a function of perceived requirements. Consequently, all too often systems are retrofitted to the intended user. Proposed here is thus the conduct of requirements analyses using some or all of the techniques described above in order to properly select the "right" dialogue technique, and input and output devices.

2.1.3.1 User Manuals. So often a system is either partially completed and then transferred or completely developed without requirements analyses and then transferred. Almost always the User's Manual is an afterthought conceived and constructed in a vacuum relative to the intended user. First and foremost, User's Manuals should never be system capabilities driven: they should always be requirements (of the intended user) driven. Secondly, they should be animated, that is, heavily steeped in graphical/visual explanations and illustrations--again in the context of user requirements. Third, technical detail, while

pleasing to the scientist/developer, should be in the Appendix or non-existent. Fourth, they should be iterative and flexible. Fifth, they should be short. Finally, they should be modular.

2.1.3.2 Transfer Logistics. Following a successful demonstration it is necessary to assess prospects for and difficulties associated with actual transfer. This generally involves assessments regarding the transferee's hardware and software capabilities. Such assessments enable CSM/DDF personnel to tailor and/or modify the system(s) to be transferred in a expeditious manner. CSM thus prefers to, when feasible and at the direction of DARPA/CTD, conduct on-site analysis of the transferee's capabilities and affect transfer accordingly. Every effort is then made to accommodate his capabilities and other requirements necessary to affect the transfer, and special attention can be devoted to maintaining the professionalism connected with the particular transfer in question and the transfer process in general.

In an effort to avoid a transfer short fall, and the loss of valuable feedback, CSM prefers to initiate a follow-through transfer procedure whereby a user will not find himself abandoned after an on-site visit and an initial tutorial session. Instead, CSM maintains a detailed transfer record and interacts with the transferee on a scheduled and ad hoc basis.

Finally, obviously the success of any transfer is dependent upon the quality and quantity of system(s) documentation made available to the transferee. As part of its follow-through strategy, CSM provides initial documentation as well as updated (systems and data) documentation.

# 3.0 THE DEVELOPMENT, DEMONSTRATION, AND DOCUMENTATION OF ADVANCED $C^2$ COMPUTER-BASED SYSTEMS

## 3.1 Computer-Based Development

### 3.1.1 Hardware

As noted in section 1.1.2.1 the selection of hardware is of critical importance to the success of a computer-based system. CSM has implemented a threefold strategy for the selection of the most appropriate hardware. First, and as a function of the design analysis, CSM can recommend a specific hardware configuration. Secondly, and with cost effectiveness in mind, CSM will canvass both the in-house and other DARPA/CTD-owned hardware in an effort to assure timely and appropriate selection(s). Too often it is the case that hardware selection is made solely on the basis of what is immediately available to the researcher. While this may stimulate rapid software production, it often creates sets of chain-reaction problems. Accordingly, before production begins, CSM attempts to match system design requirements with available (though not necessarily "at hand") hardware. Finally, CSM attempts periodically to assess the "state-of-the-art" and the "state-of-the-art likely to be", of computer hardware technology in order to avoid short-sighted implementation. In this regard, periodic memoranda are provided to DARPA/CTD focusing upon new and innovative developments in computer hardware technology.

### 3.1.2 Software

Software selection is also a function of design requirements, availability, and technological advances. CSM recommends to "developers" that computer-based systems be implemented with languages that adhere to these three criteria.

First, as a function of the design analysis, CSM will recommend a specific software language compiler or interpreter to be used. Secondly, and with cost effectiveness in mind, CSM will evaluate languages available on-line versus those generally available but not resident on the DDF computer system library. As in the hardware selection process so to is it the case that the software language selection is made solely on the basis of what is immediately available to the researcher. While this may stimulate rapid software production, it often compounds transfer and conversion problems later on in the life of the software system. CSM, therefore, before coding begins, attempts to match system design requirements with available (though--again--not necessarily "at hand") languages.

When possible, the "consistent strategy" also applies to the selection of computer software. Too often, systems have been implemented in different languages because they were either available or preferred. Indeed, there are instances of multi-language system implementation. CSM thus

attempts to assure intra- and cross-system software compatibility.

CSM also attempts to develop and rigorously apply accepted standardized (but not inhibiting) programming practices, such as structured, top-down, modularized, and flow-oriented coding techniques.

### 3.1.3 Coordination

Through its internal staff, a judicious set of reports and memoranda CSM attempts to coordinate among the user and transfer community the development of research and applications software and data. Every effort is thus made to acquaint the DDF community with each other's work. It is hoped that through an elaborate set of communications practices, the DARPA/CTD research program will not be retarded by the "not invented here" and "re-invent the wheel" syndromes.

### 3.1.4 Time-Sharing and Stand-Alone Computer Support

In order to support the design, development, demonstration, and documentation of computer-based systems, CSM operates a multiple DEC PDP 11/70 timesharing system and other real-time or stand alone microprocessor-based computer systems as so directed by DARPA/CTD.

Computer-based systems development function must be

prepared to support the three possible paths of advanced research. First, CSM responds to development requirements arising from the typical user which reflect a "light" computer requirement research effort. The second path has requirements which demonstrate a set of special needs, reflecting a research effort relying more heavily upon the computer than is typical. These special needs may range from real-time applications to very large memory specifications, storage, or heavy central processing unit (CPU) utilization. This need can be satisfied by scheduling dedicated time on the secondary "heavy research" system. Third, a very special applications requirement may arise that cannot be easily categorized as either heavy or light computer-based research, and thereby not be assigned to the primary or secondary system. The solution to this requirement often is found in the selection of a microprocessor based-computer system. More and more frequently, the hardware selection process defines a clear need to adopt the portability and stand-alone capability that a microprocessor can provide.

The areas in which CSM concentrates primary development support ensure that the hardware and software configurations are providing maximum utility to the user community are:

- Operation of the time-sharing service;
- Providing user-community programming and software support;

- Expanding the facility library to track the needs of the users and DARPA/CTD;

- Providing engineering support to all areas inside and outside the DDF; and

- Providing new product support.

Maintenance and operation of three GFE DEC PDP 11/70 mini-computer systems are as follows: the operating hours are 24 hours a day, seven days a week with one shift (prime working hours) of attended operation. The remaining time is unattended. The only exceptions to this schedule include normal Preventive Maintenance (PM) conducted by the manufacturer, Emergency Maintenance (EM) downtime, and at the direction of the director of DARPA/CTD.

Performance of quarterly hardware evaluation brings to the surface all difficulties in "throughput" as related to hardware errors, failures, and configuration flaws. Information from these evaluations is then reported to the manufacturer, CSM management, or DARPA/CTD with recommendations for corrective action.

Execution of software programs to gather data on the usage and other accountable resources utilized by the user community are conducted on a monthly basis; and the performance of daily, weekly and monthly system dumps assures a current back-up of the users files.

CSM also stands ready to support the software needs of the microprocessor users, as they are defined. Related to all software activities, CSM attempts to take positive action in the performance of:

- assistence to the user community to correct coding errors ("bugs");
- assisting or advising users in the selection and use of all DDF software packages and languages;
- assistence to users in the proper development of their applications software coding design and testing;
- responsiveness in the creation of new software utilities by addressing the problems of new or modified software; and the
- performance of ad hoc software performance evaluations to determine the "state-of-the-software". Reports describing the performance and existing problems will be distributed in such a manner as to insure prompt attention to the repair or replacement of that software routine.

Because of the ever heightening activity in computer-based systems research the DDF library has increased importance in its role as a repository of completed works. Research conducted now will hopefully become the foundations upon which future breakthroughs will occur.

Especially because of the size and complexity of the CSM/DDF operation, it is becoming increasingly more appropriate to become less reliant upon the services of non-DDF personnel. Therefore, CSM has included an <u>engineering support function</u> in the development function. The very first effort in this response area was the construction of a new and environmentally sound computer room. Since nearly all of the existing DDF inventory is government furnished equipment (GFE) it is essential that it be given the undivided attention of specially qualified personnel. Other than the initial phase of computer-room construction, CSM has conducted the following activities under engineering support:

- Electronic component parts repair;
- Facility layouts, and design in the areas of drawings, construction, mechanical and electrical plans;
- Terminal repairs and renovations that require few spare parts inventories;
- Power consumption level monitoring and conservation;
- Special wiring, cabling, and inter-device/system connections;
- Special small projects design, (e.g., small relays for voice actuated system, or control podium for demo room); and
- Communications equipment support and performance of quarterly status evaluations.

Finally, new products are always appearing in the market place. It is the function of computer-based systems development to be aware and educated about the capabilities, weaknesses, cost and availability of these new devices. CSM believes that the DEC PDP 11/70 may very well be the "dinosaur of the 1980s". We feel that it is very important to be perpared for an 11/70 replacement. By doing this we will facilitate a smooth "down-grade" toward what the technology future has already predicted. More specifically, CSM attempts as a function of the role which it proposes to play for DARPA/CTD through the DDF, examine the full range of implications for the development of computer-based systems which have already been suggested by the current revolution in very large scale integration (VLSI).

### 3.2 Demonstration

If one is at all familiar with the DDF, one realizes the importance of the role of demonstration. A new and exciting era has dawned in computer audio-visual techniques via the first production version of the spatial data base management system (SDMS). More importantly, the SDMS is supported at the DDF. This has opeaned up entire new vistas of demonstrational techniques.

#### 3.2.1 Opposing Cultures

In order to assure that applied computer-based systems

are tailored during the design and development stages, CSM intercedes (only) at the direction of DARPA/CTD during these stages to evaluate systems development against perceived user needs and requirements. At a general level this results in the application of known and standardized user-oriented programming techniques, including ease of input, menu selection, and effective display, among other features. At the more specific level, attention is devoted to the particular intended use of the system to be demonstrated.

### 3.2.2 Demonstration Staging

Extending from our concern about the inherently opposing cultures is a concern for the physical environment in which the demonstration takes place. Accordingly, very special care is always taken to create a positive professional demonstration milieu. At the most basic level, this care is manifest in an overt marketing strategy, from which a user is "handled" in a manner appropriate to his rank and responsibility.

### 3.2.3 Premature Demonstrations

In an effort to avoid at all cost a demonstration failure resulting from the premature release of a computer-based system, CSM has developed and applied a set of demonstration "readiness criteria." Specifically, the criteria are relevant

to the "state-of-the-system" in the following ways: "Has the system been thoroughly checked for all software errors and bugs?" "Has a coherent demonstration sequence been per-tested?" "Can the user operate the system easily by himself?" "Is the system flexible enough (and, where appropriate, are the data sets extensive enough) to permit a flexible demonstration?" "Have back-up technical and non-technical materials been adequately prepared?" "Have likely questions been anticipated?" "Has the back-up system been thoroughly tested?" "Has supporting (carry-away) documentation (sample-output and user's manuals) been prepared?" (See APPENDIX A)

### 3.3 Documentation

If all of this is to be routinized, we must begin to document all of the problems and progress made toward the development and transfer of useful $C^2$ computer-based aids of all kinds. CSM has prepared and submitted a plan for such documentation. It appears in APPENDIX B.

## 4.0 THE DESIGN AND DEVELOPMENT OF $C^2$ DECISION AND FORECASTING SYSTEMS

Thus far we have uncovered a great deal of information relevant to the design, development, demonstration, and transfer of advanced $C^2$ systems. Our intention was to examine the whole process rather than focus only upon display devices, specific software languages, or data base management systems. However, in FY80 specific research was conducted in the design and development ($D^2$) of computer based $C^2$ decision and forecasting systems ($C^2$D&FS), as suggested below.

### 4.1 Requirements Analysis

Requirements analyses must precede the $D^2$ of <u>all</u> $C^2$D&FSs. If such analyses cannot be conducted by DARPA personnel then contractors and/or consultants (preferably with <u>real</u> experience) should be hired to do the job. The analyses should be organizational/bureaucratic <u>and</u> substantive. The requirements analytical method should be a hybrid questionnaire/interview/ "critical incident technique" method geared to not only uncover requirements, but to determine in no uncertain terms the kind of user likely to actually <u>use</u> the system.

### 4.2 Hardware

We have already determined that for a whole host of reasons the PDP 11/70 is to remain the $C^2$D&FS's minicomputer.

However, real questions arise regarding the necessity of the PDP 11/70. Given the power of today's microcomputer systems, there are very few reasons to use the 11/70 as a transfer system (one can still argue convincingly that there are crunch applications and transfer site requirements reasons for 11/70 use, however--but not for long!).

First, we must decide upon the relative advantages and disadvantages of S-100, non-S-100 and so-called turnkey systems. S-100 systems are attractive because of the standardized connection. Our research suggests, however, that the S-100 "standard" is now being challenged by the IEEE and GFIB connections. This means that the S-100 "mix and match" capability may soon be equalled via IEEE and GFIB interfaces. Also, even though myriad options are nice, a mixed/matched system is often less reliable and physically attractive than another (non-S-100 or turnkey) system. Accordingly, our recommendation is to use a S-100 system only when the peripherals are manufactured by the same manufacturer (which ideally also manufactures the processor). Cromemco is such a manufacturer. <u>We explicitly recommend against configuring a system of many diverse parts and manufacturers</u>. While such a system may appear to be ideal on paper, in practice it simply won't work as well as a less complicated system. Finally, a "centipede" system will invite reliability and maintenance problems.

To a much lesser extent these problems exist within the non-S-100 systems, such as the Apple, Challenger, and TRS-80. These systems generally include many "detached" peripherals by the same manufacturer connected via a SS-50, IEEE, or GFIB interface. Non-S-100 systems can be acceptable if they satisfy other requirements, as presented below.

Turnkey systems are the only systems which are <u>integrated</u> in <u>design</u>, <u>function</u> and <u>appearance</u>. They also tend to offer superior compilers and interpreters, and a good deal of applications software (generally ignored by the $C^2$D&FS community). The IBM 5100, 5110, and 5120, the Tektronix 4050 and 4080 series, and the PERQ family are all turnkey systems.

Perhaps the most important mainframe characteristic is its address space. All of the microcomputer systems popular in the $C^2$D&FS community are 8 bit processors. <u>There is absolutely no reason whatsoever to continue with these systems when 16 bit microcomputer systems are available</u>.

A final extremely important consideration is storage, both main and mass. The Tektronix and IBM systems, for example, are inferior storage systems. In the future we should select systems which have serious storage capabilities.

### 4.3 Input Devices

The selection of input devices must be dependent upon the results of the requirements analysis and the type of intended user. Intuitively, it would appear that naive users would prefer spatial, touch, or speech input device types. However, our conclusion is that such input devices are not appropriate for "naive" or "commander-type" users because such devices presume interactive computing experience and familiarity which may not exist. Accordingly, keyboard, function-key and other related input devices are probably better for naive and quasi-experienced users. This, however, is not to say that programming techniques go unchanged. By and large, $C^2$D&FS contractors are in a "rut" consisting of "form-filling" and menu-selection approaches. Certainly we can do better. In any case, the use of lightpens, trackballs, mice, and the like should be rejected.

### 4.4 Display Devices

Here too requirements analyses and user profiles should dictate selection. Some general guidelines include the prudent use of graphics and hard copy capability.

Font variation and manipulation should also be possible as should techniques like blinking and coding when-- and only when--the requirements analysis and user profile suggest their appropriateness.

## 4.5 Portability

Briefly, portability means realistic transportability. In any case, the target transfer system ought to be microcomputer based; no longer should we "negotiate" for PDP 11 time and space at a transferee's site. It never works and almost complicates irreversibility our $D^2$ process.

## 4.6 Reliability

Before purchasing a microcomputer system, information should be gathered regarding the system's reliability and --much more importantly, maintenance.

## 4.7 Appearance

Appearance is subjective. However, as suggested above, multi, disconnected pieces generally are relatively unattractive.

## 4.8 Processing Speed

Speed is not a function of the microprocessor but rather overall system configuration--and software.

## 4.9 Software Languages

In our judgement, it is time that the $C^2$D&FS research community graduated from BASIC. Indeed, advanced degrees

are available in APL and especially PASCAL. It is also time that we stopped relying upon language interpretors and began to program and execute via compilers. This judgement is primarily based upon the consideration of execution speed.

### 4.10 Data Base Management Systems

We have spent a good deal of time canvassing the DBMS marketplace. We have also spent a good deal of time talking with users regarding many systems. There are a number of systems adequate for PDP 11 application.

There are also a number of microcomputer systems which, while degraded from PDP 11 system capabilities, are barely acceptable as data base managers. None of these systems are as user-oriented as $C^2$D&FS requirements would demand. Indeed, this is as true for the micro system as it is for the minicomputers (interestingly, INGRESS is not even categorized as a production DBMS). Thus, if our mission is to implement an existing DBMS and expect the interactive hand-holding present in the EWAMS, for example, then we will certainly fail. One simply can not lift a DBMS off the shelf and expect it to be user functional. Training, experience and patience are prerequisites to such use.

If, however, our intention is to hide the DBMS routines within and "behind" an applications program, then the

prospects--with significant modifications--are brighter. Indeed, there are only two ways to go: (1) select a mini (and micro) DBMS and rewrite (modify) the routines for $C^2$D&FS use or (2) identify the key DBM requirements for $C^2$D&FS applications and write a program which is treated as the $C^2$D&FS mini and micro "standard." Option 1 may make sense when a solid DBMS exists. For example, SEED and QDMS could be modified to satisfy $C^2$D&FS requirements. (However, they would have to be rewritten to some extent to function under UNIX.) On the micro side, both Cromemco and Three Rivers (PERQ) offer DBMSs that are amenable to useful modification, but level of effort estimates can not be made until a thorough analysis of the "innards" of the system can be made. Indeed, it may well be that a new system would be cheaper. Our specific recommendation at this point is to begin with existing DBMSs with a view toward modified system production. If this approach proved imprudent then we could switch to ground zero production.

The $C^2$D&FS data management requirements are not, by and large, great. Regardless of which option is selected, then, the project is doable. (Requiring contractors to adhere to the new "standard", however, could be difficult...)

## 4.11 Statistical Packages

There are also a number of PDP 11/UNIX and microcomputer

statistical "packages." We still feel strongly that SPSS (Version II) and BMDP are the best general purpose statistical packages available for PDP 11 use. In addition, there are some forecasting and simulation packages available for UNIX.

The microcomputer statistical systems are in reality disjointed routines programmed for very few statistical operations. Conversations with users indicate that storage and execution speed limitations are rampant. Even the Northwest Analytical (NWA) STATPAK is slow and limited in application. In addition, the NWA system requires micro-soft BASIC, the CP/M operating system, an 8 inch floppy disc drive, and 48K memory.

Interestingly, the Tektronix 4050 series turnkey micro-computer systems have statistical routines for numerous operations, but we have not used them for research or developmental (via source code modification) purposes.

We suspect, however, that even if such software systems were much more powerful and user-oriented that they would still not in and of themselves become the bases for $C^2D\&FS\ D^2$ because the intended product should hide the statistical routines, and not make large input demands upon the user. Not unlike data management routines, statistical routines should be invisible to the user (unless, of course, the

$C^2$D&FS is a research statistical analytical system). This realization changes our perspective somewhat when we consider the $D^2$ of generic $C^2$D&FSs insofar as the real questions have little to do with off-the-shelf statistical systems but with how efficient statistical algorithms (regardless of where they come from) can be used to process data from the DBMS routines. In other words, the task is to isolate those algorithms necessary for statistical processing and then build them into $C^2$D&FSs, regardless of whether they are adopted from existing routines or written anew.

If we isolate those statistical operations which occur regularly in $C^2$D&FS execution we see very few. Current systems implicitly or explicitly calculate modes, medians, ranges, means, frequency distributions, some low level correlations, Z-scores, and some covariance analyses. However, the operations which precede and underlie the systems themselves are much more sophisticated. This is an absolutely critical distinction: on the one hand we have statistical analytical requirements which precede or enhance $C^2$D&FS development and/or performance, while on the other we have those operations which occur during on-line system use. Our generic system should be applicable to both of these phases (as should our generic DBMS). Accordingly, we must make a series of hard decisions about which way to go vis-a-vis the two distinct $D^2$ phases as suggested below:

| Mini/Micro Antecent Analyses | Mini/Micro $C^2$D&FS On-Line Operations |
|---|---|
| • DBMS | • DBMS |
| • Statistical Routines | • Statistical Routines |

The distinction further suggests that statistical (and DBMS) requirements during the pre-$D^2$ phase may be much greater than at the system operation phase where pre-calculated files coupled woth unsophisticated statistical operations can essentially constitute "the" system. This in turn suggests that the generic mini- and microcomputer statistical (and DBMS) algorithms intended for on-line execution remain low-level, and that pre-$D^2$ algorithms be (relatively) powerful.

### 4.12 Display Coding Techniques

Competent requirements analyses should determine the nature and use of display coding techniques such as blinking, motion, depth, and color coding.

### 4.13 Interaction Mode/Dialogue Types

Until full natural language is available "standard" techniques should be used, such as menu-selection via function keys and command languages.

4.14 Summary

We have already said enough about the need for thorough requirements analyses. The microcomputer mainframe should be a 16 bit one and have ample (at least 5MB mass and 200+KB main) storage. Dual density disks are a prerequisite if floppy; an alternative hard disk is manufactured by Winchester and implemented by numerous OEMs. As of today, we recommend the PERQ system (acceptable alternatives are the Cromemco System Three and Z2-H). The PERQ is a very high speed 16 bit system (CPU) with integrated I/O controllers. It has 256KB of RAM (with a 1MB RAM option!) and a built-in 12MB rigid disk (with a 24MB option!). By comparison, these capabilities literally dwarf the now too popular Tektronix and IBM systems. Indeed, there is no fair comparison among the systems. The Cromemco (S-100) system is a Z-80 based 8 bit system with up to 512KB of RAM and 2MB of disk storage. The Cromemco Z2-H has 11MB of hard disk and 64KB of RAM (expandable to 512KB). All of these systems may be considered turnkey, if configured consistently.

Input devices are standard with the Cromemco systems, consisting of keyboards and joystick (bad)/function key (good) consoles. The PERQ system provides a (detachable) keyboard and a touch tablet (and speech output).

The Cromemco's display system is grounded primarily in

software traits and characteristics (implanted by the programmer not the manufacturer). However, since the Cromemco is an S-100 system there are many display options available. The PERQ graphics display system is a 768 point by 1024 line, bit mapped, raster scanned image on a 15" CRT. All 1024 lines are refreshed 60 times per second for flicker-free high resolution. Font can be <u>any</u> size, shape or complexity (multiple fonts are supported as well as proportionately spaced characters). In addition, the PERQ uses a <u>display window manager</u> which partitions the screen into seperate areas or "windows"," which may be moved around the screen, enlarged, or contracted <u>in two dimensions</u>, scrolled, and/or clipped under direct user control. <u>Menus can be inserted into the windows</u> (for continual display during operation) and can be as large as the entire screen or as small as a postage stamp.

Both the Cromemco and PERQ series systems are as portable (if not more so) than the Tektronix and IBM systems (including their disk drives and printers/hard copy units).

Candidly, reliability and maintenance are relatively unknown.

The PERQ is more attractive than the multi-piece Cromemco.

PERQ is extremely fast; Cromemco somewhat slower, but both are _much_, _much_ faster than the Tektronix or IBM systems.

The Cromemco Z2-H supports extended BASIC, FORTRAN IV, RATFOR, and COBOL in interpreter form. The PERQ has a _full PASCAL compiler_ thus satisfying our language/speed criterion.

The Cromemco and PERQ systems have DBMSs as part of their software support packages. Our recommendation is to work with these systems to first determine how effective they might be on-line for $C^2$D&FS operation. If they prove adequate then no modifications or new programming would be necessary; if not, then the real work would begin.

(On the minicomputer side, we recommend SEED or QDMS for research purposes.)

Statistical operations should be minimized on the PERQ; the PDP 11/70 should support more sophisticated operations _in a research mode_. In this vein, _a bona fide research system, consisting of an integrated DBMS/Statistical System, should be developed coupling SEED/QDMS with BMDP/SPSS (11)_. This would permit advanced analyses for hypothesis testing and avoid the (re-)writing of individual routines for specific projects.

On the micro side, we should adhere to the same design but not require the system to crunch large data bases or expect the micro to support sophisticated pre-$D^2$ analyses --unless we are prepared to write new micro "statpacks."

From a design perspective, we recommend a one designated micro per project arrangement. For example, TRAP could be a PERQ-based $C^2$D&FS and the XAIDS a Cromemco-based system. If projects were "assigned" a hardware/software configuration from the outset then many selection problems could be avoided. This approach would also prevent us from locking ourselves into a particular configuration enabling us to grow and evolve along with the micro market which is expanding rapidly perpetually. PDP 11/70-based systems should no longer be transfer delivery systems; rather, they should be used as research and "inventory" systems. For "delivery" purposes a set of designated micro system DBMS/Statistical Routines should be adopted, modified and/or written for the designated system and standardized.

Potential designated (powerful 8 and 16 bit) systems abound the marketplace. We have zeroed in upon PERQ and Cromemco because today they are probably the best systems available for across the board $C^2$D&FS use. But there may indeed be $C^2$D&FSs which could be implemented nicely on other systems. Provided these systems made sense from a require-

ments and hardware/software standpoints, then they might very well be appropriate.

A final capability undiscussed thusfar is networking. The PERQ's networking capabilities, for example, suggest a new kind of $C^2$D&FS. User's of the PERQ "station" can access data and programs which run on another (at 10MBits per second via a single coaxial cable using standard cable TV technology). Accordingly, one can imagine a distributed system swapping data and programs to accommodate sharing and shared decision-making and forecasting. Alternatively, a single full blown PERQ station could house several large data sets and supply other PERQ stations with different applications programs. The possible networked configurations are endless.

## 5.0 CONCLUSION

All of the above constitutes CSM's work in FY80 in the design, development, demonstration and transfer areas. In FY81, CSM hopes to continue this work especially as it relates to advanced communications, teleconferencing and robotics systems design and development.

## 6.0 FOOTNOTES

[1] See H. Rudy Ramsey and Michael E. Atwood, "Human Factors in Computer Systems: A Review of the Literature," SAI, Incorporated, September, 1979 for a larger discussion of what appears in Section 3.0 of this report.

## 7.0 REFERENCES


Ramsey, H. Rudy and Michael E. Atwood. Human Factors In Computer Systems: A Review of the Literature. Englewood, Colorado: Science Applications, Inc., September, 1979.

Wittmeyer, James F., III, DARPA/CTD/CSM/DDF Quarterly Technical Reports.

# APPENDIX A

# READINESS STATUS REPORTING FOR THE DEMONSTRATION OF ADVANCED $C^2$ COMPUTER-BASED SYSTEMS

MONTHLY SOFTWARE DEMONSTRATION AND TRANSFER READINESS REPORT

SUMMARY

Date: February, 1980

| Software Systems: | Demonstratable | Transferable |
|---|---|---|
| Early Warning System | Yes | No* |
| Executive Aids | Yes | No* |
| TRAP | No* | No* |
| OPINT | No* | No* |
| EVAL | No* | No* |
| RAM | No* | No* |
| SCORE | No* | No* |
| DECISION | No* | No* |
| Group Decision Aid | No* | No* |
| STEAMER | No* | No* |
| PLATO | No* | No* |
| PRESS | No* | No* |
| AIS | No* | No* |
| Computer Generated Maps | No* | No* |
| Duncan/Job Estimator | No* | No* |
| Ultra-Rapid Reader | No* | No* |

* See attached elaboration.

MONTHLY SOFTWARE DEMONSTRATION AND TRANSFER READINESS REPORT

ELABORATION

Date: February, 1980

| System | Status |
| --- | --- |
| Early Warning System | Demonstratable primarily because of continued familiarity with system; no back up slides; no draft system specification or functional description; no internal documentation. |
| Executive Aids | No draft system specification, functional description, or internal documentation; no back-up demonstration slides; no continued training on capabilities or evolution of system(s). |
| TRAP | Unclassified version not supported by training, viewgraphs, reports, internal documention, system specification, or functional description. |
| OPINT, EVAL, RAM, SCORE, DECISION | No software delivered; no other supporting documention except DDI's documention which is generic and (probably) peripheral to software. |
| Group Decision Aid | No software documention of any kind; no reports; no viewgraphs. |
| STEAMER | No documention of any kind. |
| PLATO | No documention of any kind. |
| PRESS | Minimum software documention; no slides; no reports; cursory user's "guide". |

| System | Status |
| --- | --- |
| AIS | Undelivered. |
| Computer-Generated Maps | No documention of any kind. |
| Duncan/Job Estimator | No documention of any kind. |
| Ultra-Rapid Reader | No documention of any kind except for a cursory user's manual. |

**Special Note: Obviously DDF personnel can--and have--given demonstrations of some of the above listed undemonstratable software systems without the requisite back-up materials. However, as we hope is clear, CSM will not always be able to give proper demonstrations without the necessary back-up and training. Transfer is much more difficult without good documention. We believe that totally effective demonstrations can only be given when all materials and serious training has been delivered.

# APPENDIX B

# A PLAN FOR THE DOCUMENTATION OF DARPA/CTD-SUPPORTED RESEARCH

# A PLAN FOR THE DOCUMENTATION OF DARPA/CTD-SUPPORTED RESEARCH

James F. Wittmeyer, III

Computer Systems Management, Inc.
1300 Wilson Boulevard
Suite 102
Arlington, Virginia 22209

## 1.0 INTRODUCTION

Over the years DARPA/CTD has supported numerous research projects in the $C^2$ information, decision, forecasting, training, and combat readiness/effectiveness areas. Much of this support has been devoted to the development, testing, and transfer (to the Services and the Intelligence Community) of computer-based systems and computer aids while other support has been devoted to more basic research necessary for the development of such systems and aids. Since much of this research is cumulative in nature, and since it is conducted in divergent fields and disciplines, it is necessary to establish procedures for acquiring, storing, documenting, and distributing the information, reports, software, and hardware connected with the research. This plan thus presents a number of ideas and suggestions for gathering, processing, and distributing information important to the chronology and progress of DARPA/CTD, ideas and suggestions which will be implemented, modified, or dismissed at the direction of DARPA/CTD.

# 2.0 APPROACH

## 2.1 Problem Scope

DARPA/CTD generated information takes many forms, including:

- Written Technical Reports;
- Written Technical Memoranda;
- Filmed Technical Reports;
- Filmed Technical Memoranda;
- Written Demonstration Scenarios and "Sample" Output;
- Filmed Demonstrations;
- Minicomputer Software;
- Microcomputer Software;
- Seminar/Workshop/Conference Proceedings;
- Minicomputer Hardware Configurations;
- Microcomputer Hardware Configurations;
- Other Hardware;
- Miscellaneous Briefing Materials; and
- Funded Technical Proposals.

All of this information could easily overtake one's organizational resources. Since DARPA/CTD is not structured to maintain such information beyond its own purposes, CSM here proposes to assist DARPA/CTD with this information management problems.

## 2.2 Proposed Solution

### 2.2.1 Information Organization

With DARPA/CTD approval, CSM proposes to organize the information according to the current CTD program thrusts cross-referenced by CTD contractor. This will facilitate the organization, retrieval, and distribution of information in a timely and efficient manner. In addition, fixed categories will be set up in an effort to maintain consistent documentation procedures, as suggested below.

Program: $C^2$ Information Systems

| Contractor / Information | MIT | CCA | BBN | ITC | Percept | DDI | ESL |
|---|---|---|---|---|---|---|---|
| Written Technical Reports | | | | | | | |
| Written Technical Memoranda | | | | | | | |
| Filmed Technical Reports | | | | | | | |
| Filmed Technical Memoranda | | | | | | | |
| Written Demo Scenarios & "Sample" Output | | | | | | | |
| Filmed Demonstrations | | | | | | | |
| Minicomputer Software | | | | | | | |
| Minicomputer Software | | | | | | | |
| Seminar/Workshop/ Conference Proceedings | | | | | | | |
| Minicomputer Hardware Configurations | | | | | | | |
| Microcomputer Hardware Configurations | | | | | | | |
| Other Hardware | | | | | | | |
| Miscellaneous Briefing Materials | | | | | | | |
| Funded Technical Proposals | | | | | | | |

It is recommended that the matrices for all of the CTD programs/information/contractors be computerized into a master inventory which would guide the collection and maintenance, and distribution of materials. INGRES could easily be used to cross-reference the inventories for information types, contractor, and/or program or, just as easily, a small utility program could be written.

### 2.2.2 Information Collection

There is no question that CSM will need the assistance of DARPA/CTD to gather the suggested information. Fortunately, the CTD/CSM/DDF already has a good number of reports and other documents gathered over the past three to four months. These reports have been cataloged and organized for easy library-like retrieval. They have not been computerized, however. Also, the annual CTD Hardware/Software Inventory has yielded a good deal of information about hardware and software configurations. Finally, the DDF equipment inventory constitutes another excellent source of information.

In order to gather and organize information beyond DDF's immediate resources, the CTD contractor community will have to be contacted and be willing to cooperate. In the past, contractor response has been at the 60% level. This response rate will have to increase if we are to assemble a representative collection of materials; certainly "encouragement" from CTD will greatly improve our chances for success.

Finally, for organization purposes we feel that a test case involving either a single contractor (like MIT or Perceptronics) or one CTD program should be attempted first so we can identify the issues and problems relevant to the collection and organization process.

## 3.0 TIMETABLE

The target date for the computer-based working information management system has realistically been set for March 31, 1980. However, in order to meet this date we will require prompt response from the contractors and our consultants. If we are able to receive the information from the contractors (including lists, reports, films, and the like) within sixty (60) days, then we can maintain the schedule. If not, then our schedule will slide accordingly. Further, with good cooperation, we should be able to get a single contractor or program system running by January 31, 1980.

## 4.0 CONCLUSION

All of this is subject to DARPA/CTD review. We are anxious to build the proposed system and invite comments, criticisms, and suggestions.